



User-Centric Power Optimization in Smart Home Environments

By

Saurabh Gupta

Under the supervision of

Dr. Arun Balaji Buduru, IIIT Delhi

Dr. Ponnurangam Kumaraguru, IIIT Hyderabad

Indraprastha Institute of Information Technology Delhi

July, 2022

©Indraprastha Institute of Information Technology Delhi
(IIITD), New Delhi, 2020



User-Centric Power Optimization in Smart Home Environments

By

Saurabh Gupta

Submitted

in partial fulfillment of the requirements for the degree of
Master of Technology

to

Indraprastha Institute of Information Technology Delhi

July, 2022

Certificate

This is to certify that the thesis titled “**User-Centric Power Optimization in Smart Home Environments**” being submitted by **Saurabh Gupta** to Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

July 2022

Dr. Arun Balaji Buduru
Department of Computer Science & Engineering
Indraprastha Institute of Information Technology Delhi
New Delhi, India

Acknowledgements

I want to thank my thesis advisors Dr. Arun Balaji Buduru, IIIT Delhi, and Dr. Ponnurangam Kumaraguru, IIIT Hyderabad, for their constant support and supervision. I am immensely grateful to them for providing me with their valuable guidance and insights through the course of the thesis. I also want to thank my colleagues at the Precog Lab, IIIT Delhi, who volunteered for any support or help I needed for this research project. Finally, I want to express my gratitude to my family for providing me with unfailing support and continuous encouragement through researching and writing this thesis.

July 2020

Saurabh Gupta
Computer Science and Engineering
Indraprastha Institute of Information Technology Delhi
New Delhi, India

Abstract

Rapid advancements in the Internet of Things (IoT) have facilitated efficient deployments of smart environment solutions for specific user requirements. With the increase in the number of IoT devices, it has become difficult for the user to control or operate every individual smart device into achieving some desired goal like optimized power consumption, scheduled appliance running time, etc. Smart homes require every device inside them to be connected with each other at all times, which leads to a lot of power wastage on a daily basis. As the devices inside a smart home increase, it becomes difficult for the user to control or operate every individual device optimally. Therefore, users generally rely on power management systems for such optimization but often are not satisfied with the results. In this work, we present a novel multi-objective reinforcement learning framework with two-fold objectives of minimizing power consumption and maximizing user satisfaction. The framework explores the trade-off between the two objectives and converges to a better power management policy when both objectives are considered while finding an optimal policy. We experiment on real-world smart home data, and show that the multi-objective approaches: i) establish trade-off between the two objectives, ii) achieve better combined user satisfaction and power consumption than single-objective approaches. We also show that the devices that are used regularly and have several fluctuations in device modes at regular intervals should be targeted for optimization, and the experiments on data from other smart homes fetch similar results, hence ensuring transfer-ability of the proposed framework.

Contents

Certificate	i
Acknowledgements	ii
Abstract	iii
List of Figures	vii
List of Tables	viii
1 Introduction	1
2 Background	5
2.1 Power Optimization	5
2.2 Traditional Reinforcement Learning	8
2.3 Multi-objective Reinforcement Learning	10
3 Problem Formulation	12
3.1 Environment	13
3.2 Power Controller (Agent)	15
3.3 Reward	16
3.3.1 Minimizing Power Consumption	16
3.3.2 Maximizing User Satisfaction	16
3.3.3 Overall reward	17
3.4 Evaluation	18

4	Solutions	19
4.1	Single Policy Single Objective	20
4.1.1	Power Consumption Minimization	20
4.1.2	User Satisfaction Maximization	20
4.2	Single Policy Multi Objective	21
4.2.1	Equal weights	21
4.2.2	Weighted Sum	21
5	Environment Setup	22
5.1	Smart* Data Set for Sustainability	22
5.2	Designing the Environment	23
5.2.1	Data Processing	23
5.2.2	Clustering to assign the modes of operation for each device	24
6	Experiments	26
6.1	Single Policy Single Objective	27
6.2	Single Policy Multi-Objective	28
7	Results	29
7.1	Average Power	29
7.2	Average Number of Clashes	30
7.3	Appliance-wise clashes	31
7.3.1	Transferability on other smart homes consumption data	33
8	Conclusion	34
	Bibliography	37

List of Figures

1.1	Electricity consumption in the US from 1950-2020. Evidently, the electricity consumption is highest for residential purposes, and the trend is not changing anytime soon.	2
2.1	Architecture of Traditional Reinforcement Learning.	8
2.2	Architecture of Multi-Objective Reinforcement Learning.	10
3.1	Illustration of Power controller. To formulate the problem, we create an environment with several states in it defined by a vector of the device states at any given time. We define a reward which consists of two components: i) power optimization, and ii) user satisfaction. Given an initial state, the agent (i.e. the power controller) can take an action to change the environment's state and calculate a reward. The next action then depends on the reward from the previous step. Detailed explanation of each component is available in the text. . .	14
5.1	Clustering results for Duct Heater from smart home A: Silhouette analysis for K-means clustering with $k = 3$ for Duct Heater.	25
5.2	Clustering results for Duct Heater from smart home A: Distribution of consumption values of Duct Heater with $k = 3$	25
7.1	Average Power vs No. of Episodes for different approaches.	29
7.2	Average Number of Clashes vs No. of Episodes using different approaches.	30
7.3	Appliance wise clash rate for Furnace.	31

7.4	Appliance wise clash rate for Heater.	31
7.5	Appliance wise clash rate for Lights	32
7.6	Appliance wise clash rate for Refrigerator.	32
7.7	Appliance wise clash rate for Washer.	32
7.8	Average reward and Clashes with Weighted Sum approach smart homes A, B, and C.	33

List of Tables

6.1	Hyper-parameter values for Single Policy Single-Objective Q-learning	27
6.2	Hyper-parameter values for Single Policy Multi-Objective Q-learning	28

Chapter 1

Introduction

The amount of power consumption in households (residential) is among the top three in world electricity consumption [IEA, 2019], and is ever increasing with the increase in demand of smart homes and IoT (Internet of Things) devices. According to the United States Department of Energy (DoE), the average household consumes 90 million units of power a year, and much of that power is wasted [Center, 2019]. Habits like leaving lights on when we leave rooms, forgetting to turn off televisions or computers when not in use, etc., are major reasons behind such wastage [Constellation, 2019]. Therefore, there is a need for power controllers that can take actions like turning devices on and off, or changing devices' modes of operation on behalf of users to achieve a goal like optimized consumption.

The increasing involvement of smart devices in our lives has necessitated us to come up with better coordination and management strategies for increased efficiencies and Quality of Service (QoS). Many organizations are investing in extensive research to come up with strategies to make utilization of energy as efficient as possible. "Smart Homes" or "Smart Devices" is one such area that makes use of Information & Communication Technologies for finding solutions to such environmental issues. With the help of embedded intelligence, such devices can target the preferences of a user during everyday life. However, cogency of such results is critical [Chisik, 2011, Kempton and Montgomery, 1982]. With the installation of smart meters along with these devices at homes, we can find effective ways to pro-

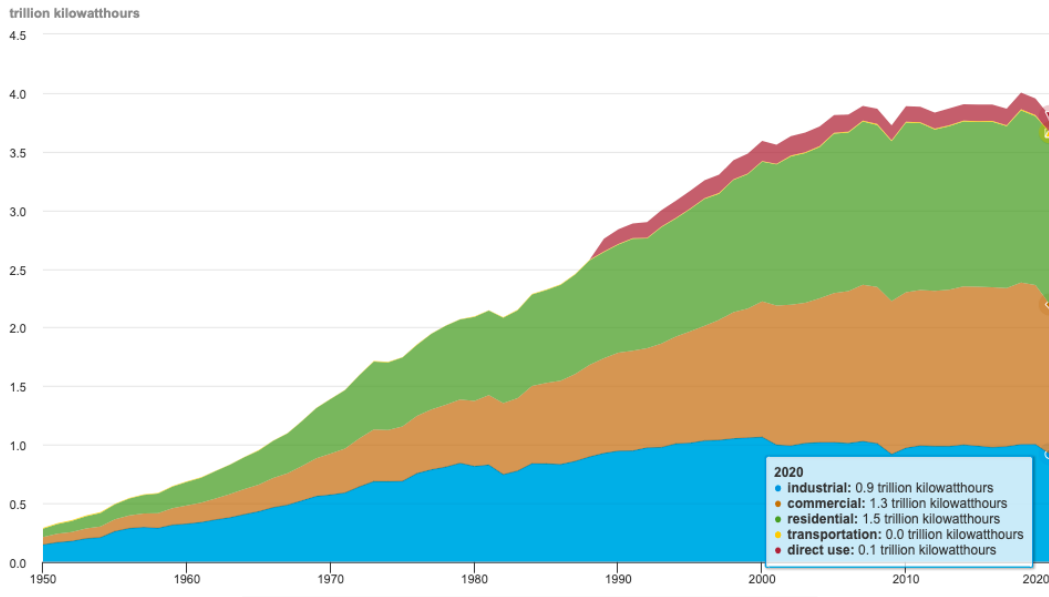


Figure 1.1: Electricity consumption in the US from 1950-2020. Evidently, the electricity consumption is highest for residential purposes, and the trend is not changing anytime soon.

vide users with a visualization of their energy consumption pattern in a way they can comprehend [Herrmann et al., 2018]. Furthermore, we can even find ways to make intelligent systems that can recommend users or act on their behalf, based on their past power consumption behavior, to achieve the goal of minimizing electricity wastage. The objective here needs to be to prevent any unnecessary consumption of energy, preserving the comfort and productivity of the user.

In smart homes, a vast number of heterogeneous appliances, sensors, and actuators inter-operate and provide context information, which in turn, together with user preferences, are used to effectuate a value-added functionality dynamically. A smart home needs to be able to analyse the actions of its occupant, taking into account the context information, to proactively recognize the occupant’s activity to conserve energy. Therefore, while creating such systems, one must proceed in a way such that the intelligent planner’s policies in promoting efficient energy usage in households must take into account the user’s choices and behavior for utmost user satisfaction [Goulden et al., 2014].

In the past, researchers have used traditional reinforcement learning for several power optimization tasks. For example, [Tan et al., 2009] proposed a model-free constrained RL approach for online power management. [Shen et al., 2013] presented another similar algorithm that requires no prior information of the workload and dynamically adapts to the environment to achieve autonomous power management. [Tesauro et al., 2008] proposed an RL based technique that performs simultaneous online management of both performance and power consumption. The authors applied RL in a realistic laboratory testbed to find the optimal policy. None of these techniques towards power optimization are used for smart home power management, and they do not consider user satisfaction while finding optimal policies.

However, power management in a smart home is a problem that needs to solve two tasks with different rewards simultaneously: minimize power consumption and maximize user satisfaction. It is important for a power controller to consider user preferences as well, i.e., the goal of minimal power consumption must be achieved but not at the expense of user satisfaction. The scenario can be formulated as a MORL problem where sequential decision making is required with multiple objectives. We propose a novel multi-objective reinforcement learning (MORL) approach for power management inside a smart home with two objectives: minimize power consumption and maximize user satisfaction. In a MORL problem, an action on the environment results in multiple rewards. The agent (power controller) learns an optimal policy from these rewards using a variation of Q-learning [Watkins and Dayan, 1992]. Since the objectives are contrasting, there is a trade-off between the two, and based on their importance, optimization priorities are set. We use an overall reward function to incorporate these optimization priorities, which is a weighted sum of the two rewards R_E representing power consumption, and R_U representing user satisfaction. We specifically focus on the weighted-sum method [Kim and De Weck, 2006] for multi-objective optimization and compare the results with single objective strategies.

We evaluate our proposed methods on the Smart* data set for sustainability [Barker et al., 2012]. The data samples include device-level real-world power consumption values in several smart homes, named as A, B, C, ..., H recorded at every 30 minutes. We show the effectiveness of our approach on data from smart home

A, and demonstrate transfer-ability of experiments on smart homes B and C. We use Q learning with individual objectives (single policy single objective approaches) as a baseline reference to compare the proposed single policy multi-objective approaches. We also define a metric “clash rate” for evaluating user satisfaction in the predicted policy at each episode.

Chapter 2

Background

2.1 Power Optimization

The increase in computing power in constrained environments have allowed researchers to experiment and come up with new and better approaches to solve some critical problems using the power of artificial intelligence. Hence, there have been some approaches proposed earlier which either have been mainly theoretical in nature or have focused only on one specific part of this problem statement. We present a brief discussion on these methodologies adopted by researchers in the past.

In works that study the behavioural clustering of devices, Adika et.al. present a method where appliances are clustered together based on their hourly energy consumption data [Adika and Wang, 2014]. Then a “time of use” probability distribution is made, and each cluster is given a schedule, making this a job scheduling problem which is solved using dynamic programming. The challenge here is to predict the correct cost of energy for the next hour. Consequently, the scheduling assumes that the error in predicting the cost is minimum. However, the approach does not focus on any method to suggest the user ways to cut down on excessive power consumption.

In yet another approach, the authors introduce a generic way of creating a Peer to Peer (P2P) overlay network using Hydra middle ware [Jahn et al., 2010, Eisenhauer et al., 2010]. The authors talk about collecting and displaying energy consump-

tion data using several data visualisation techniques. The approach does not have intelligent monitoring or recommendation for the users, which makes us ask the question of whether the data visualisation effects the user's understanding of energy consumption or not [Herrmann et al., 2018].

Wei et. al. show another general architecture for designing the energy consumption monitoring and energy-saving management system that are IoT based [Wei and Li, 2011]. According to this work, IoT infrastructure has three levels: the bottom-most level collects data from different sensors, the middle level which is the network layer talks about data transmission, and the topmost level processes the collected and transferred data using cloud computing and fuzzy pattern recognition techniques. A 3-component architecture was also introduced to cater to an intelligent household lighting system for efficient energy consumption, including user's context information such that the system's behaviour conforms to user satisfaction [Byun et al., 2013]. Although this approach talks about an intelligent planner, it is limited to the lighting systems and uses a static algorithm (minimum light intensity control) to control the lights. Hence, the approach is not efficient when it comes to dynamic environments where user behaviour is prone to change or in a real-life setting where a set of diverse electrical equipment or appliances are present.

Yau et. al. base their technique on Markov Decision Process (MDP), assuming there are three entities [Yau and Buduru, 2014]. Firstly, the manufacturer provides the set of device states, secondly a SESP (Smart Environment Service Provider) who analyses the state information and lastly, the MDP planning algorithm which makes a policy of states and actions along with a user who receives these actions suggested by the MDP and acts accordingly in the given environment. The approach assumes a Central Module which collects all the information and does the heavy lifting, in turn, solving the problem of having a constrained environment. The approach involves manual human intervention, firstly as the manufacturer who supplies information on the devices and then in the role of SESP who performs analysis on the data. Also, The results are shown over simulation, and the approach is never used on a real data-set.

The methodology used by Jahn et. al. and Byun et. al. makes us under-

stand the major approaches for collecting data from appliances and hence, helps us move in the direction of intelligent planning [Jahn et al., 2010, Byun et al., 2013]. The architecture proposed by Wei et. al. is a general way to move from the stage of data collection to intelligent planning [Wei and Li, 2011]. There are existing approaches that are perfectly capable of achieving optimal energy consumption, taking the assumption that the user behaviour is not dynamic, and the price prediction has minimum error [Adika and Wang, 2014, Byun et al., 2013]. The approach used by Yau et. al. uses RL and hence, is capable of dealing with a change in user behaviour [Yau and Buduru, 2014]. However, the role of entities like manufacturer and SESP can be automated. In this work, we find the quality of states based on the user behaviour, which keeps on updating itself to improve with time. The work presented in this work is an advancement of the approach discussed in [Yau and Buduru, 2014].

In this section, we discuss traditional reinforcement learning with Q-learning, an algorithm widely used to solve traditional RL problems. Then we introduce the concepts of multi-objective reinforcement learning (MORL) and how it differs from the traditional RL.

2.2 Traditional Reinforcement Learning

Traditional reinforcement learning [Sutton and Barto, 2018] mimics the natural learning style of trial-and-error by interacting with an environment (static or dynamic) and receiving feedback based on an action. The components of reinforcement learning are:

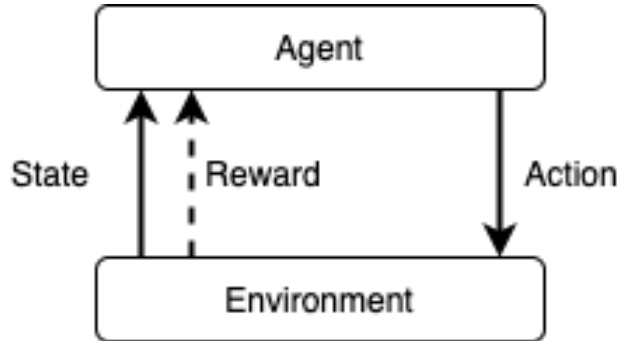


Figure 2.1: Architecture of Traditional Reinforcement Learning.

- An Agent;
- A finite state space S ;
- A set of available actions for the agent A ;
- A reward function $R : S \times A \rightarrow R$.

The agent's objective is to maximize its average long-term reward. It is achieved by learning a policy π , which is a mapping between the states and the actions. In our problem, one goal is to minimize the power consumption of a smart home, and the other is to maximize user satisfaction. But, in a traditional reinforcement learning setting, the two goals are independent. An agent can either minimize power consumption, or it can maximize user satisfaction.

Q-learning [Watkins and Dayan, 1992] is a widely known algorithm used to solve sequential decision-making RL problems. In each step, on the successful execution of every action a , the environment yields a reward R , which indicates the value of a state transition. The issued reward can be positive or negative. The agent

keeps a value function $Q^\pi(s, a)$ for each state-action pair. Learning to act in the environment will make the agent choose actions to maximize long-term rewards. Based on this value function, the agent decides its immediate action. The Q-value for each state-action pair is initially chosen during the problem formulation, and later, it is updated with each taken action and its issued reward. The value function is given by the following Bellman equation:

$$Q^\pi(s, a) = R(s, a) + \gamma \max_a Q^\pi(s', a) \quad (2.1)$$

where $R(s, a)$ is the reward issued after taking action a in state s , s' is the successive state of s , and γ is the discount factor used due to the different influences of future rewards on the present value.

The optimal state-action value function is defined as:

$$Q^*(s, a) = \max_\pi Q^\pi(s, a) \quad (2.2)$$

When $Q^*(s, a)$ is obtained, the optimal policy π^* can be computed by:

$$\pi^*(s) = \operatorname{argmax}_a Q^*(s, a)$$

2.3 Multi-objective Reinforcement Learning

Reinforcement learning is a machine learning paradigm that helps with sequential decision making under several uncertainties and aims to achieve a single long-term objective. However, due to the complex requirements of real-world control systems, often times, there are two (or more) conflicting objectives. For example, in our case of smart home power management system the controller has two goals: i) to minimize energy consumption of the smart home, ii) to maximize user comfort by moving to states preferred by the user. In reinforcement learning, problems of this nature having more than one conflicting objectives are called multi-objective reinforcement learning problems (MORL).

MORL is different from traditional RL in that there are two or more objectives to be optimized simultaneously by the learning agent. [Liu et al., 2014] provides an architecture for a MORL problem, where reward is provided for the learning agent at each step. In MORL (Figure 2.2), there are N objectives and $r_i(1 \leq i \leq N)$ is the i^{th} reward signal provided by the environment. The architecture illustrates a single agent that has to find an optimal policy for a set of multiple objectives simultaneously. The objectives can be conflicting, as in our case, or they can be independent as well.

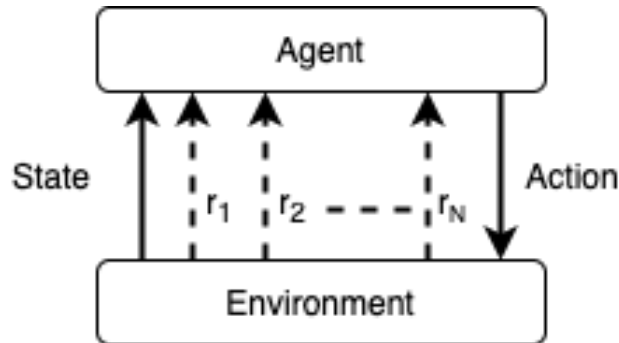


Figure 2.2: Architecture of Multi-Objective Reinforcement Learning.

For each objective i ($1 \leq i \leq N$) and a stationary policy π , there is a corresponding state-action value function Q_i^π , which satisfies equation 2.1.

Let the combined value function for MORL is:

$$MQ^\pi(s, a) = [Q_1^\pi(s, a), Q_2^\pi(s, a), \dots, Q_N^\pi(s, a)]$$

where $MQ^\pi(s, a)$ is a vector and it also satisfies the Bellman equation (2.1). Then the optimal state-action function will be given as:

$$MQ^*(s, a) = \max_{\pi} MQ^\pi(s, a) \tag{2.3}$$

and the optimal policy π^* can be obtained by:

$$\pi^*(s) = \operatorname{argmax}_a MQ^*(s, a) \tag{2.4}$$

MORL is a combination of multi-objective optimization methods and RL techniques to solve sequential decision making problems with multiple conflicting objectives. We will justify why we formulate smart home power management as a MORL problem in the next section.

Chapter 3

Problem Formulation

The case of a smart home power management system is a multi-objective problem with two objectives, viz., minimizing power consumption and maximizing user satisfaction. Ideally, a controller will try to reduce the power consumption as much as it can, given an optimization goal. The trivial solution for the controller will be to turn off all the devices that operate in the smart home. However, this state might not be desirable by the user. Therefore, it is important for a controller to consider user preferences as well. Hence, the goal of minimal power consumption must be achieved by establishing a trade-off with user satisfaction, and not at its expense.

Based on the importance of an objective function, optimization priorities must be ensured while designing the policies. After appropriately expressing the preferences, we have to design an efficient algorithm that can solve the sequential decision making problems based on observed state transition data.

3.1 Environment

Smart homes usually have smart meters to measure the power consumption for each device operating within it. The power consumption values for every device is independent, and take a fixed number of discrete values. This is because each device operates only in a fixed number of modes and their power consumption in a specific mode remains the same. For example, a simple furnace has only two modes, ON and OFF. In OFF mode, it consumes no power while in ON mode it consumes x (say) units of power. We are assuming that the consumption x remains the same and no degradation of device happens over time, hence causing more energy consumption. Similarly, a washing machine can have three modes of operation, viz, standby, washer and dryer. Let's assume a smart home has N devices. A state in an environment is a vector of energy consumption values (in whatever device modes they are in) of these N devices, as depicted by blocks s and s' in Figure 3.1.

Let's assume the number of device modes each one operates in is given by a set $D = \{nD_1, nD_2, \dots, nD_N\}$

$$\text{Total number of states in state - space} = \prod_{i=1}^N nD_i$$

Increasing the number of devices, or just their modes of operation can lead to state-space explosion. Therefore, in our techniques we choose devices selectively and use data processing to avoid state-space explosion.

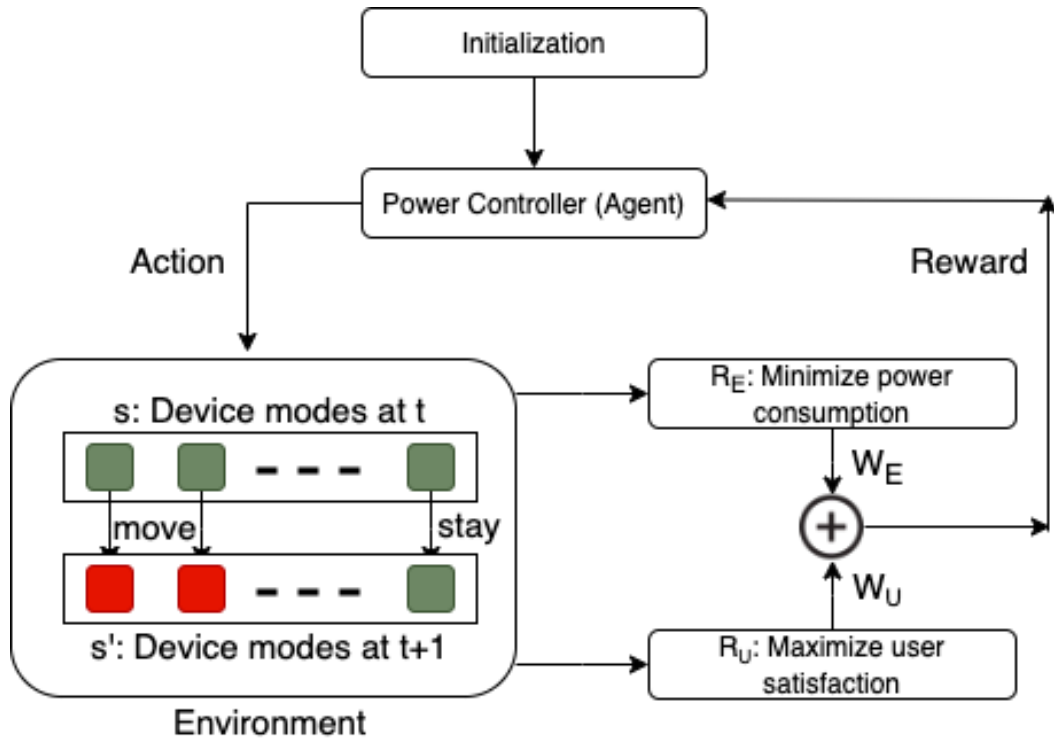


Figure 3.1: Illustration of Power controller. To formulate the problem, we create an environment with several states in it defined by a vector of the device states at any given time. We define a reward which consists of two components: i) power optimization, and ii) user satisfaction. Given an initial state, the agent (i.e. the power controller) can take an action to change the environment's state and calculate a reward. The next action then depends on the reward from the previous step. Detailed explanation of each component is available in the text.

3.2 Power Controller (Agent)

The Agent is a power controller that can change the mode of operation of any of the N devices, consequently changing the energy consumption value. For example, a power controller can turn off the furnace, if it is on, or switch the washing machine to dryer mode from some other mode. However, an agent can also choose not to do anything. Therefore, the Agent can perform either of the two actions, i.e., *MOVE* or *STAY*, on each device in a state s , to move the environment to a state s' . For example, in Figure 3.1, the controller changes the mode of operation of device 1 and device 2, from *green* to *red* by *MOVE* action, and chooses to keep the N th device in its current state by *STAY* action.

3.3 Reward

Whenever the proposed agent takes an action on the environment, a reward is calculated on the basis of the state chosen by the controller and the ground truth state from the Smart House Dataset. Since there are two distinct objectives, we formulate the reward functions to incorporate the power consumption and user satisfaction. Every update of the state-action value function (Equation 2.1) is dependent on the reward. Therefore, by integrating the optimizations in reward function, the agent learns the trade-off between optimization priorities for an optimal state. First we introduce both rewards separately and then we combine them to form a single reward, as shown in Figure 3.1.

3.3.1 Minimizing Power Consumption

Let's say the power consumption in the predicted and ground truth state is $P_{s'}$, and P_s respectively. The reward, R_E is given as:

$$R_E = \frac{\sum_d -(P_{s'} - P_s)}{D} \quad (3.1)$$

which is the average difference of power consumed by the D devices between predicted state and the desired state. As the agent always tries to maximize the reward, thus we negate the sum in order to achieve state which consumes less power than what the user had chosen. By negating, the state with the least electricity consumption becomes goal state for the power controller.

3.3.2 Maximizing User Satisfaction

To model user behavior, we compute the Euclidean distance between the predicted state and the ground truth state. The reward

$$R_U = \frac{\sum_d |s' - s|}{D} \quad (3.2)$$

where $d \in D$ devices.

3.3.3 Overall reward

We take a weighted combination of both the rewards, R_E (Equation 3.1) and R_U (Equation 3.2), and define overall reward as:

$$R = W_E * R_E + W_U * R_U \quad (3.3)$$

where W_E and W_U are the weights to manipulate the optimization priorities of the two objectives. These weights are treated as hyper parameters during experimentation.

3.4 Evaluation

The evaluation of power controller’s performance is two-fold due to the multi-objective nature of the optimization problem. The reward R_E represents negation of power consumption, therefore, a policy with more positive R_E value is desired. Hence, as we increase the number of iterations, the value of R_E should increase.

Similarly, the reward R_U represents the likelihood that the next state predicted by the controller, S_U^{pred} , matches to the next state that user prefers, S_U^{real} . However, to evaluate R_U , we introduce a term called “clash rate” to get a device level view of clashes. To calculate clash rate:

$$clash\ rate = \sum(S_U^{real} == S_U^{pred}) \quad (3.4)$$

where “==” is an element wise comparison that assigns 1 if values do not match and 0 otherwise, and returns an array with 1’s and 0’s.

For example, let us say the user wants next state to be $S_U^{real} = [D_1, D_2, D_3, D_3, D_1]$, where D_i s represent the device modes at this state. Now, the controller takes an action on the environment to change its state to $S_U^{pred} = [D_2, D_1, D_3, D_1, D_1]$. The clash rate in this case is 3, as the device modes at index 0, 1, and 3 do not match (assume the vectors are indexed starting from 0). As we increase the iterations to train the power controller more, the clash rate should decrease.

Chapter 4

Solutions

MORL approaches can be divided into two groups based on the number of policies to be learned [Vamplew et al., 2011]: single policy and multiple policy approaches. In our case, the objectives are contrasting, and the availability of data allows us to create a sufficiently good representation of the environment. Therefore, we focus on a single policy approach to solve it.

The aim of single policy approaches is to obtain the best policy which satisfies the optimization priorities as set by the designer, or defined by the application domain. Therefore, based on varying optimization priorities we implemented four variations of a single policy algorithm to find an optimal policy for our two-fold objectives of minimum power consumption and maximum user satisfaction. A single policy approach to solve MORL problems is to formalize an objective function $TQ(s, a)$, which can represent overall preferences in optimization. The approach is very similar to Q-learning with a few modifications, as shown in Algorithm 1. The objective function $TQ(s, a)$ is given as the summation of Q-values for all the objectives, and is given as:

$$TQ(s, a) = \sum_{i=1}^N Q_i(s, a) \quad (4.1)$$

As discussed in Section 3.3, we incorporate the optimization priorities using weights W_E and W_U in the reward function. Since $Q(s, a)$ is dependent on the reward

Algorithm 1 Single Policy Approach to solve MORL

```
1: K: the maximum number of episodes
2: N: the number of objectives
3: Initialize  $TQ(s, a)$ ;
4: Initialize  $Q_i(s, a), \forall(i < N)$ ;
5: for each episode  $j$  ranging from 1 to  $K$  do
6:   Fetch  $s_0^j, s_1^j$  from samples;
7:   Choose  $a$  using  $TQ(s, a)$  policy using  $\epsilon$ -greedy approach;
8:   Take action  $a, s'$ ;
9:   Compute reward  $[r_1, r_2, \dots, r_N]$  based on  $s_1^j$ , and  $s'$ ;
10:  for  $i = 1, 2, \dots, N$  do
11:     $Q_i(s, a) \leftarrow Q_i(s, a) + \alpha[r_i + \gamma \max_{a'} Q_i(s', a') - Q_i(s, a)]$ ;
12:  end for
13:  Compute  $TQ(s, a)$ ;
14:   $s \leftarrow s'$ ;
15: end for
```

function, and $TQ(s, a)$ on $Q(s, a)$, any change in the weight values in equation 3.3 will result in a change of values in $TQ(s, a)$. They are defined as:

4.1 Single Policy Single Objective

As a baseline reference, we implement single policy approach with single objectives. Recall equation 3.3, the overall reward is defined as the sum of two rewards, one for minimizing power consumption and other for maximizing user satisfaction. Therefore, for single policy with one objective taken at a time has two cases:

4.1.1 Power Consumption Minimization

To implement this, we give 100% optimization priority to power consumption, and set W_E and W_U to 1 and 0, respectively in equation 3.3.

4.1.2 User Satisfaction Maximization

To implement this, we give 100% optimization priority to user satisfaction, and set W_E and W_U to 0 and 1, respectively in equation 3.3.

4.2 Single Policy Multi Objective

The goal of the power controller is to achieve a multi-objective optimization. Therefore, we consider two cases:

4.2.1 Equal weights

The case where both objectives are equally as important, and the power controller tries to optimize both. Both W_E and W_U are set to 1. Based on the policy calculated by the power controller, the action with the maximum summed values is chosen to be executed.

4.2.2 Weighted Sum

The weighted sum approach is proven to be effective with multiple objectives in the past. [Ngai and Yung, 2011] used it to combine seven vehicle overlapping objectives, and [Zeng et al., 2010] used it with a combination of three objectives, viz., degree of the crowd in an elevator, the waiting time, and the number of start-ends. The approach modifies equation 4.1 as:

$$TQ(s, a) = \sum_{i=1}^N W_i Q_i(s, a)$$

In our case, the W_i s are W_E and W_U , and we experiment with different values of both to get the best results.

Chapter 5

Environment Setup

We evaluate the proposed solutions in Section 4 using the Smart* data set for sustainability [Barker et al., 2012]. As a baseline reference, we consider Q-learning with single objective of power consumption minimization and user satisfaction maximization. We plot the reward and clashes for all four proposed algorithms to contrast the results. In this section, we first briefly explain the data set, then the environment design, and finally the experiments and results.

5.1 Smart* Data Set for Sustainability

The data set includes real power consumption readings of multiple devices such as furnace, fridge, washing machine, etc., inside smart homes collected over regular intervals of 30 minutes. Each device has sensors attached to them to record the power consumption after regular time intervals inside seven smart homes¹. For our experiments, we used data from smart home A collected over a period of three years. We also use data from smart home B, and C to show transferability of the framework. Note that data from each smart home is similar in nature. The only difference is in the type and the number of devices used to collect data.

¹<http://traces.cs.umass.edu/index.php/Smart/Smart>

5.2 Designing the Environment

The data set has power consumption values from more than 20 devices for each smart home. We have considered only 5 devices from a smart home: furnace, washing machine, fridge, heater, and kitchen lights. The reasons to do so are:

- In a real world scenario, a user does not want the controller to operate on all of the devices in their smart home.
- Formalizing an optimization problem with only a top few devices with maximum power consumption is more realistic and helpful than taking all the possible devices and constraints into consideration.
- For simplicity of experimentation.

5.2.1 Data Processing

In our data set, the power consumption reading for each device took many distinct unique values. For example, the power consumption values for Furnace has 17,000 unique entries, and that of Fridge is 16,000. However, a lot of these values are very close and differ only at $4^{th}/5^{th}$ decimal place representing a data collection glitch. Since we chose 5 devices, a **state** in this environment is represented by a vector of size $[D_1^1, \dots, D_i^k, \dots, D_2^5]_{1 \times 5}$ where D_i^k represents the device D^k in mode i , and its value is given as the power consumption by device k in mode D_i .

The size of the state space is the cross product of all the unique values taken by each device. Therefore by this convention, if we consider only the furnace and the fridge, the size of state space will be 272 million ($16,000 \times 17,000$). With such a big state space, the problem becomes very complex to solve, and therefore, to avoid the state space explosion, we cluster the energy consumption values of each device separately to find a fixed **number of modes of operation** for each device. Intuitively, in real-life, a furnace cannot have 17,000 modes of operation. Therefore, finding device modes with clustering seems to be a fair assumption to make.

5.2.2 Clustering to assign the modes of operation for each device

We wanted to find cluster centers of power consumption values for each device individually, which can represent different modes of operation. The modes of operation can be readily available from manufacturer’s end but they might not be ideal for our case. For example, suppose a washing machine consumes x power in standby mode, y in wash mode and z in dry mode, and the values, y and z are very close. The manufacturer can say that modes y and z are different, but we have similar readings for the two states, and hence, does not affect our objective. Therefore, we cluster the readings such that each device mode represents a significant amount of change in power consumption from one mode to another. The clustering helps us reduce the state space to a very good extent.

First we performed silhouette analysis [Rousseeuw, 1987] to find the optimal number of clusters for each device. We vary k , the number of clusters from 2 to 6 assuming it is rare that a device has more than 6 modes of operation. Figure 5.1 shows the silhouette plot for various clusters using Duct Heater’s electricity consumption data for $k = 3$. The clusters are well formed with coefficient value more than the threshold, as can be seen in Figure 5.2. The plot is shown for $k = 3$ as it yielded the best silhouette score. Similar experiments are performed with remaining 4 devices.

After clustering, the optimal number of clusters for the chosen devices, Furnace, Washing Machine, Fridge, Duct Heater, and Kitchen Lights is 2, 3, 3, 3, and 5, respectively. Clustering reduces the size of state space from 272 million+ ($16,000 \times 17,000$) to just 270 ($2 \times 3 \times 3 \times 3 \times 5$) preventing the state space explosion.

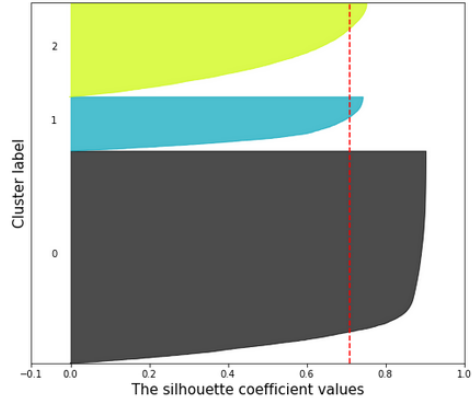


Figure 5.1: Clustering results for Duct Heater from smart home A: Silhouette analysis for K-means clustering with $k = 3$ for Duct Heater.

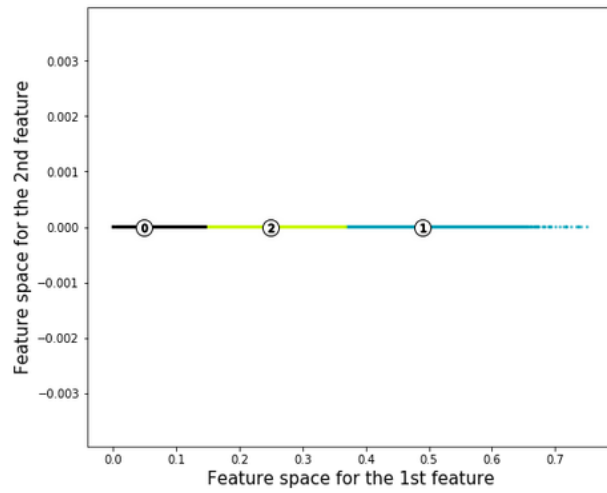


Figure 5.2: Clustering results for Duct Heater from smart home A: Distribution of consumption values of Duct Heater with $k = 3$

Chapter 6

Experiments

The first objective is to minimize the total power consumption, and the second objective is to maximize user satisfaction. Our algorithm takes into account 4 hyper-parameters: learning rate (α), discount factor (γ), exploration rate (ϵ), reward prioritization weights W_E and W_U . As a baseline, we use Algorithm 1 with single objectives. Note that if we run the algorithm with a single objective, it becomes the traditional Q-learning algorithm. The clash rate as defined in Equation 3.4 will be maximum for single policy with power consumption minimization objective, and minimum with user satisfaction maximization objective. However, with multi-objectives, the clash rate should be between the two. The overall reward is given as the weighted sum, therefore, reward will be maximum for multi-objective approach. We implemented the solutions discussed in Section 4 as:

6.1 Single Policy Single Objective

The overall reward has two weighted terms, W_E and W_U representing power consumption and user satisfaction, respectively. For the first set of experiments, we focus only on optimizing a single objective by initialising W_E and W_U as $(1, 0)$ for power consumption minimization objective, and $(0, 1)$ for user satisfaction maximization objective. Hence, in single policy single objective Q-learning formulation, our agent only receives the reward R_E in the former case and the reward R_U in the latter.

We experimented with more than 100 combinations of α and γ with $\alpha, \gamma \in (0, 1]$ to find the best hyperparameters. The agent calculates average total reward R and the *clashrate* for every combination of our hyperparameters over a total of 463 unique states episodes learned over 300 epochs. We decayed the value of ϵ by a factor of 1.4 every 20 epochs. The set of parameters which gives us the highest average reward and least number of clashes is chosen. The hyperparameters shown in Table 6.1 achieve the best results when our aim is to minimize the average number of clashes to meet each objective individually.

Table 6.1: Hyper-parameter values for Single Policy Single-Objective Q-learning

<i>Objective/Hyper-parameters</i>	Power Consumption Minimization	User Satisfaction Maximization
α	0.4	0.9
γ	0.1	0.05
ϵ	0.1	0.1
W_E	1	0
W_U	0	1

6.2 Single Policy Multi-Objective

We divide the experiments for multi-objective approaches into two approaches as discussed in Section 4: Equal Weights and Weighted-Sum. As shown in Line 11 of Algorithm 1, the update function for the Q-values is different than the normal Q-learning formulation. The equation for the Q-value update is given as:

$$Q_i(s, a) \leftarrow \alpha(r_i + \gamma \max_{a'} Q_i(s', a') - Q_i(s, a)) \quad (6.1)$$

For equal weights approach, the weights W_E and W_U have been assigned the same value of 1 representing equal priority for both objectives. For Weighted-Sum approach, we perform experiments by taking approximately 2,300 combinations of α , γ, ϵ , W_E and W_U with their values within the range (0,1]. The best hyperparameters for the multi-objective approaches are listed in the Table 6.2.

Table 6.2: Hyper-parameter values for Single Policy Multi-Objective Q-learning

<i>Approach/Hyper-parameters</i>	Equal Weights Approach	Weighted-Sum Approach
α	0.9	0.9
γ	0.1	0.1
ϵ	0.05	0.07
W_E	1	0.3
W_U	1	0.1

Chapter 7

Results

7.1 Average Power

To compare the four algorithms proposed to find an optimal policy, we ran them for equal number of epochs using the best hyperparameters obtained for each. Each epoch has 463 training steps and 450 validation steps, we plotted the average power for each epoch for comparison. Figure 7.1 shows the average reward for each algorithm.

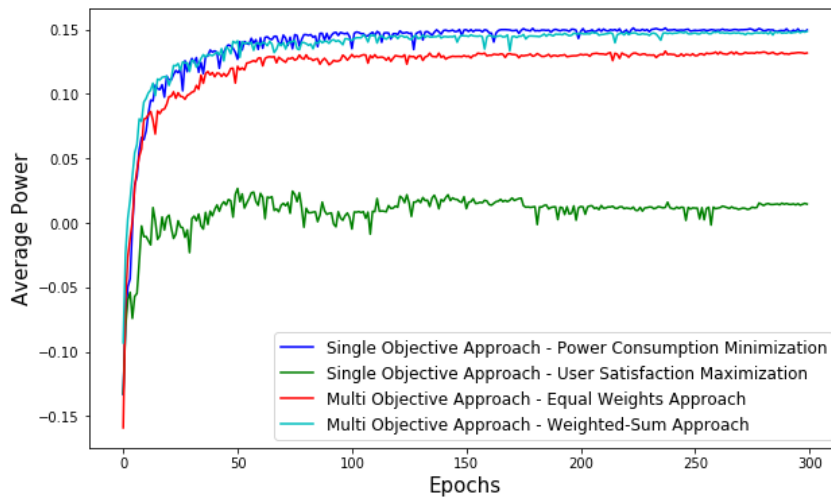


Figure 7.1: Average Power vs No. of Episodes for different approaches.

The average power is maximum for power satisfaction minimization because the

policy is getting reward based on the predicted state and user’s next state, and it is lowest for user satisfaction maximization due to the fact that if we move from a high power state to a low power state, it will hurt user’s satisfaction, which is indeed the desired behavior. The plots for multi-objective approaches always end up between the two single objective ones, representing the trade-off between the two contrasting objectives.

7.2 Average Number of Clashes

Figure 7.2 shows the combined clash rate for all four algorithms. The experimental parameters are kept same as the previous section. Note that for power consumption minimization the clash rate is highest because no weightage is given to user satisfaction. If we deploy an agent with such policies, the user will get agitated and they will try to override agent’s actions, rendering it useless. While on the other hand, an agent with user satisfaction maximization policies will not be helpful in optimizing power consumption. However, if we look at clash rates for multi-objective techniques, they lie between the two single objective approaches, and this clash rate can be adjusted using weights based on user preferences.

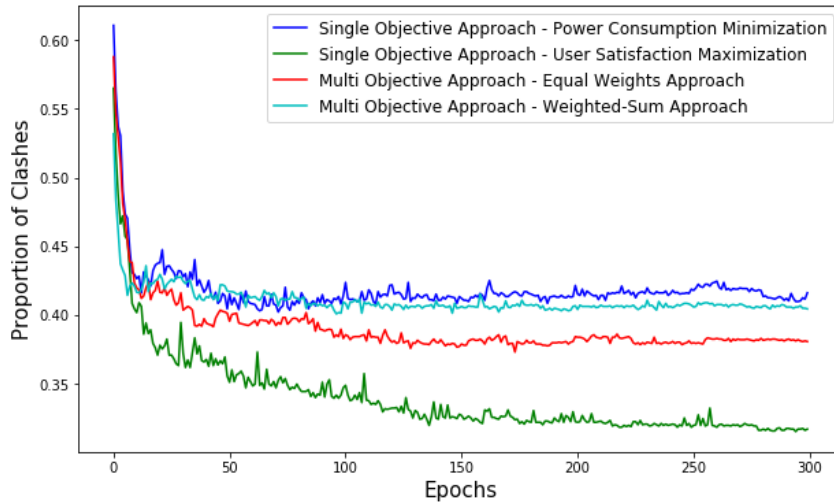


Figure 7.2: Average Number of Clashes vs No. of Episodes using different approaches.

7.3 Appliance-wise clashes

We calculate the average number of clashes for each of the five appliances and plot them separately to see the behavior of proposed approaches. Figures 7.3, 7.4, 7.5, 7.6, 7.7 show the clash rate for each device. The experimental parameters are kept same as Section 7.1.

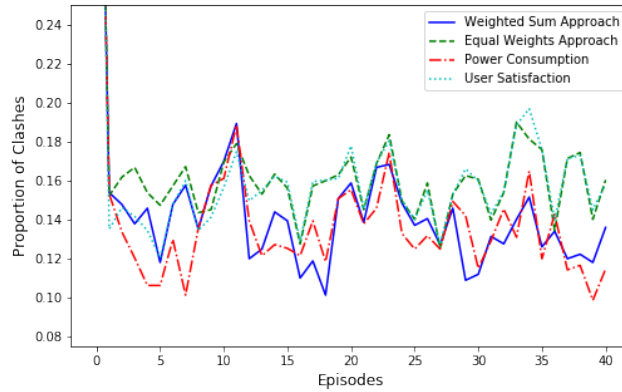


Figure 7.3: Appliance wise clash rate for Furnace.

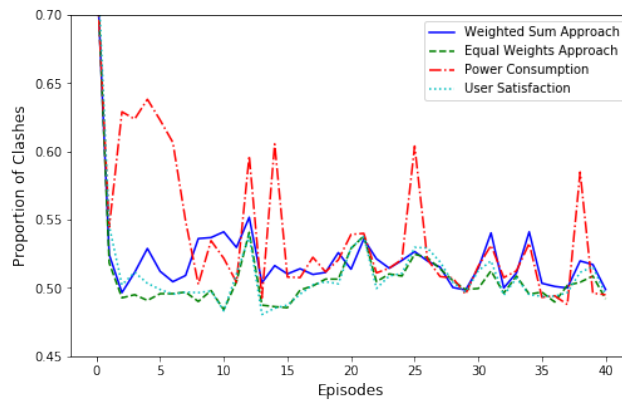


Figure 7.4: Appliance wise clash rate for Heater.

The results for appliances with three device modes is consistent with the overall results with two exceptions of furnace (two device modes) and kitchen lights (5 device modes).

For furnace, the power consumption minimization approach does not behave as expected. The reason could be the irregular usage and collection of data, as a



Figure 7.5: Appliance wise clash rate for Lights

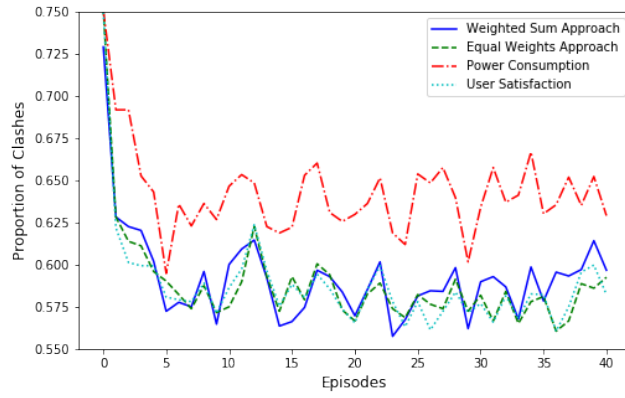


Figure 7.6: Appliance wise clash rate for Refrigerator.

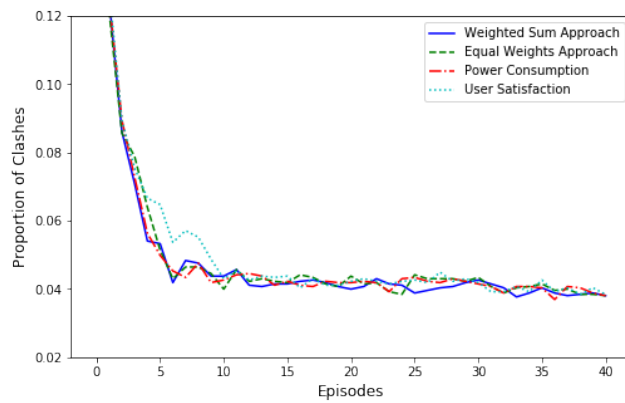


Figure 7.7: Appliance wise clash rate for Washer.

furnace is used only during colder seasons and the data we used for experiments is collected over a span of three years. For lights, all algorithms fetch nearly same results. The reason can be because lights are used for prolonged times, and there are not many fluctuations in lights' modes of operation. Therefore, the clash rate coincides for user satisfaction, power consumption, and a combination of the two. Hence, Furnace and Lights have very little to contribute to the overall optimization. The results, therefore, suggest that devices that are used regularly and with several fluctuations in device modes at regular intervals should be targeted for optimization.

7.3.1 Transferability on other smart homes consumption data

To show that the proposed framework can be applied to power consumption data of multiple smart homes, we choose the best algorithm (weighted sum approach) and run it for smart homes B, and C from the same Smart* data discussed in Section 5.1.

Figure 7.8 shows that the rewards increase and the clash rate decreases with increase in number of episodes. Figure 7.8a and Figure 7.8b show that the behavior is similar on all three smart homes data.

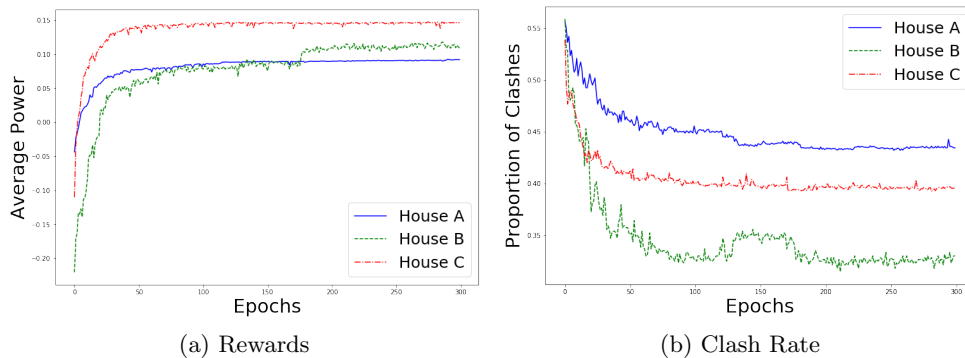


Figure 7.8: Average reward and Clashes with Weighted Sum approach smart homes A, B, and C.

Chapter 8

Conclusion

In this work, we present a novel multi-objective reinforcement learning technique for power consumption optimization with contrasting objectives of minimizing power consumption and maximizing user satisfaction. We show that both objectives, when considered together, achieve the best optimal policy. Our experimental results show that the proposed multi-objective techniques establish a trade-off between the two objectives. The optimal policy achieves better user satisfaction than power optimization policies and achieves better power consumption than user maximization policies. We show that the devices used regularly in smart homes should be the ones targeted for such optimization purposes. Finally, we also show that the experiments can be performed with other smart home data set to achieve similar results.

Bibliography

- [Adika and Wang, 2014] Adika, C. O. and Wang, L. (2014). Autonomous appliance scheduling for household energy management. *IEEE Transactions on Smart Grid*, 5(2):673–682.
- [Barker et al., 2012] Barker, S., Mishra, A., Irwin, D., Cecchet, E., Shenoy, P., Albrecht, J., et al. (2012). Smart*: An open data set and tools for enabling research in sustainable homes. *SustKDD, August*, 111(112):108.
- [Byun et al., 2013] Byun, J., Hong, I., Lee, B., and Park, S. (2013). Intelligent household led lighting system considering energy efficiency and user satisfaction. *IEEE Transactions on Consumer Electronics*, 59(1):70–76.
- [Center, 2019] Center, R. P. S. (2019). Energy data facts.
- [Chisik, 2011] Chisik, Y. (2011). An image of electricity: Towards an understanding of how people perceive electricity. In Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., and Winckler, M., editors, *Human-Computer Interaction – INTERACT 2011*, pages 100–117, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Constellation, 2019] Constellation (2019). Energy-saving strategies for smart homes.
- [Eisenhauer et al., 2010] Eisenhauer, M., Rosengren, P., and Antolin, P. (2010). Hydra: A development platform for integrating wireless devices and sensors into ambient intelligence systems. In Giusto, D., Iera, A., Morabito, G., and Atzori, L., editors, *The Internet of Things*, pages 367–373, New York, NY. Springer New York.

- [Goulden et al., 2014] Goulden, M., Bedwell, B., Rennick-Egglestone, S., Rodden, T., and Spence, A. (2014). Smart grids, smart users? the role of the user in demand side management. *Energy Research & Social Science*, 2:21 – 29.
- [Herrmann et al., 2018] Herrmann, M. R., Brumby, D. P., Oreszczyn, T., and Gilbert, X. M. P. (2018). Does data visualization affect users’ understanding of electricity consumption? *Building Research & Information*, 46(3):238–250.
- [IEA, 2019] IEA (2019). World electricity final consumption by sector, 1974-2017.
- [Jahn et al., 2010] Jahn, M., Jentsch, M., Prause, C. R., Pramudianto, F., Al-Akkad, A., and Reiners, R. (2010). The energy aware smart home. In *2010 5th International Conference on Future Information Technology*, pages 1–8.
- [Kempton and Montgomery, 1982] Kempton, W. and Montgomery, L. (1982). Folk quantification of energy. *Energy*, 7(10):817 – 827.
- [Kim and De Weck, 2006] Kim, I. Y. and De Weck, O. (2006). Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation. *Structural and multidisciplinary optimization*, 31(2):105–116.
- [Liu et al., 2014] Liu, C., Xu, X., and Hu, D. (2014). Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3):385–398.
- [Ngai and Yung, 2011] Ngai, D. C. K. and Yung, N. H. C. (2011). A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):509–522.
- [Rousseeuw, 1987] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65.
- [Shen et al., 2013] Shen, H., Tan, Y., Lu, J., Wu, Q., and Qiu, Q. (2013). Achieving autonomous power management using reinforcement learning. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 18(2):1–32.

- [Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- [Tan et al., 2009] Tan, Y., Liu, W., and Qiu, Q. (2009). Adaptive power management using reinforcement learning. In *2009 IEEE/ACM International Conference on Computer-Aided Design-Digest of Technical Papers*, pages 461–467. IEEE.
- [Tesauro et al., 2008] Tesauro, G., Das, R., Chan, H., Kephart, J., Levine, D., Rawson, F., and Lefurgy, C. (2008). Managing power consumption and performance of computing systems using reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1497–1504.
- [Vamplew et al., 2011] Vamplew, P., Dazeley, R., Berry, A., Issabekov, R., and Dekker, E. (2011). Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine learning*, 84(1-2):51–80.
- [Watkins and Dayan, 1992] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.
- [Wei and Li, 2011] Wei, C. and Li, Y. (2011). Design of energy consumption monitoring and energy-saving management system of intelligent building based on the internet of things. In *2011 International Conference on Electronics, Communications and Control (ICECC)*, pages 3650–3652.
- [Yau and Buduru, 2014] Yau, S. S. and Buduru, A. B. (2014). Intelligent planning for developing mobile iot applications using cloud systems. In *2014 IEEE International Conference on Mobile Services*, pages 55–62.
- [Zeng et al., 2010] Zeng, F., Zong, Q., Sun, Z., and Dou, L. (2010). Self-adaptive multi-objective optimization method design based on agent reinforcement learning for elevator group control systems. In *2010 8th World Congress on Intelligent Control and Automation*, pages 2577–2582. IEEE.